Seminar Nasional Teknologi dan Rekayasa (SENTRA) 2019
ISSN (Cetak) 2527-6042
eISSN (Online) 2527-6050

APLIKASI BERBASIS WEBSITE SEBAGAI INTERFACE UNTUK OVER THE AIR UPDATE FIRMWARE PADA PERANGKAKAT IOT

Supriyanto[1], Wahyu Andhyka Kusuma*[2],Mahar Faiqurahman[3]
[1,2,3]Universitas Muhammadiyah Malang / Malang

Kontak Person:
Wahyu Andhyka Kusuma
Jalan Raya Tlogomas No. 246, Tlogomas, 65144,
(0341) 463513
E-mail: kusuma.wahyu.a@gmail.com

Abstrak

Perangkat IoT yang diimplementasi pada berbagai tempat, dianggap sistem yang tidak perlu mengubah requirement dan fungsinya. Padahal kenyataanya perangkat IoT yang berjalan pasti akan berubah dapat berupa behavior, parameter yang terkait komunkasi dengan sistem lain atau penguna, memperbaiki kesalahan, dan dapat berupa update keamanan. Perubahan dilakukan dengan cara update firmware perangkat sistem langsung dengan mengunakan usb to micro usb menghubungkan ke perangkat IoTdan laptop melalui komunikasi serial. Hal ini tidak masalah jika perangkat IoT hanya 2 atau 3 perangkat namun akan menjadi masalah jika perangkat Io Tada diberbagai tempat seluruh penjuruh dunia.Solusi yang dilakukan adalah dengan memanfaatkan protokol MQTT untuk melakukan over the air update firmware pada perangkat IoT yang sudah terhubung ke internet, dan mengunakan aplikasi berbasis website sebagai interface penguna untuk upload file firmware perangkat IoT lalu di PUBLISH ke perangkat IoT.

Kata kunci: *OTA, FIRMWARE, MQTT, IOT, website*

1. Pendahuluan

Pada akhir 2013, ada 9,1 miliar unit perangkat *Internet of Things* (IoT) dengan konektifitas *Internet Protocol* dan berkomunikasi tanpa interaksi dengan manusia, *Internasional Data Corporation (IDC*) memperkirakan pertumbuhan IoT yang diterapkann tiap tahun mencapai 17.5% diperkirakan menjadi 28,1 miliar di tahun 2020 [1]. Bahkan *Cisco* mempunyai predeksi dua kali lipat lebih besar yaitu 50 miliar pada tahun 2020 [2]. Perangkat IoT yang digunakan diberbagai tempat sering dianggap sebagai sistem yang tidak perlu mengubah *requirements* dan fungsinya, namun, pada kenyataanya dimana perangkat IoT ini berjalan pasti akan berubah[3]. Perubahan ini meliputi perubahan *behavior*, parameter yang terkait komunikasi dengan sistem lain atau penguna, memperbaiki kesalahan, bisa masalah keamanan, yang dilaporkan penguna setelah perangkat IoT digunakan[3].

Berbagai perubahan perangkat IoT dapat dilakukan dengan menganti *firmware*, untuk mengganti *firmware* pada perangkat IoT harus keluar mengambil perangkat IoT, menghubungkan ke komputer, melakukan *update* dan mengembalikan perangkat IoT ke tempat. Namun, hal ini tidak dapat terus dilakukan bagi perusahaan yang memiliki perangkat IoT di berbagai tempat, seperti yang dilakukan *Chrysler* "merek mobil" pada tahun 2015 mereka dikritik karena mengirim perangkat *flashdisk* kepelangan untuk melakukan *update firmware* karena sangat rentan, *flashdisk* dapat diambil, di modifikasi dan dikirim kembali[4].

Jika sistem pada perangkat IoT sudah dapat berkomunikasi melalui antarmuka jaringan, hal ini bisa dimanfaatkan untuk menerapkan pembaruan firmware pada sistem IoT yang disebut dengan *Over The Air* (OTA)[3]. OTA di lakukan oleh *Tesla* pada tahun tahun 2016 mengirimkan pembaruan *firmware* pada mobil mereka dan konsumen dapat mengatur akan melakukan pembaruan pada saat mobil di parkir[4].

Over The Air update adalah proses memuat *firmware* pada modul ESP "perangkat *IoT*" mengunakan koneksi jaringan Wi-Fi dari pada mengunakan kabel *port serial*[5]. Secara umum isitilah OTA adalah mekanisme pengunaan *wireless* untuk mengirim data, memperbarui paket untuk pembaruan *firmware* atau perangkat lunak ke perangkat *mobile*, sehingga penguna tidak perlu pergi mengakses fisik perangkat untuk mengubah aplikasi, parameter, *firmware*, atau memperbarui *software*[6].

ISSN (Cetak) 2527-6042
eISSN (Online) 2527-6050

Over The Air Update pada perangkat IoT sudah ada di pasaran dalam produk merek *Libelium*, namun *OTA* hanya bisa dilakukan pada perangkat IoT *Libelium* melalui *File Tranfer Protocol* (FTP). Selain itu ada *particle.io* sama seperti *Libelium* hanya bisa digunakan untuk perangkat IoT yang mereka sediakan tidak bisa untuk perangkat lain. Selain dua perangkat berbayar *libelium* dan *particle.io*, OTA di sediakan oleh *espresif* melalui produk bernama *esp8266 12-E*, dapat melakukan update *firmware* melalui *Arduino ide*, web browser , dan *HTTP Server*[5].

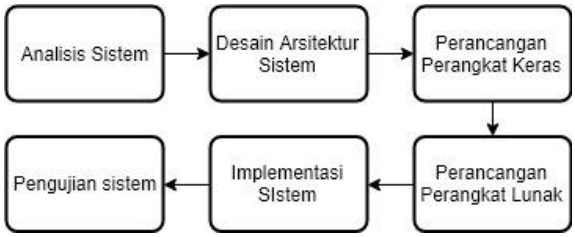
Over The Air Update dengan *esp8266* 12-E yang telah tersedia melalui *arduino ide* dan web browser memilik keterbatasan hanya bisa dilakukan dalam satu jaringan yang sama, dan HTTP Server bisa dilakukan dengan jaringan berbeda[5]. Namun pada pada penelitian pengujian sensing data suhu dan kelembapan pada penelitian sebelumnya menunjukan protokol HTTP 6 kali lebih lambat melakukan transfer data dari pada *protokol MQTT* dalam 60 detik dalam 5 kali percobaan di dapatkan rata-rata *HTTP* 934.4 data terkirim dan *MQTT* 6520.2 data terkirim[7].

Protokol MQTT adalah protokol pesan *publish/subcribe*, sangat sederhana, dan ringan, dirancang untuk perangkat yang terbatas oleh jaringan dengan *bandwidth* rendah, latensi tinggi atau tidak dapat diandalkan. Prinsip desain *protokol MQTT* adalah untuk meminimalkan *bandwidth* jaringan dan kebutuhan sumber daya perangkat, sambil berusaha memastikan kehandalan dan beberapa tingkat kepastian pengiriman data benar-benar terkirim *Quality of Service (QoS)*[8]. Contoh pengunaan protokol MQTT adalah Facebook Messenger[9] pada awal peluncuranya tahun 2011.

Berdasarkan latar belakang tersebut akan diimplementasikan *protokol MQTT* yang dapat melakukan *Over The Air update* pada perangkat *IoT*. *Over The Air update* dengan *protokol MQTT* akan digunakan untuk melakukan *update firmware*, *software* pada perangkat IoT, dari jaringan lokal atau jaringan internet, melalui media aplikasi berbasis website. Aplikasi berbasis website sebagai media *interface* penguna untuk melakukan *update file firmware* dan melakukan monitoring hasil *upload* apakah berhasil atau tidak. Protokol MQTT digunakan sebagai media pengiriman *file firmware* hasil *build* dari *Arduino IDE* berupa *file tipe bin*, yang di *publish* ke perangkat IoT yang telah melakukan s*ubscribe* pada suatu *topic*.

2. Metode Penelitian

Penerapan *over the air update firmware* mengnakan protokol MQTT dengan aplikasi berbasis website sebagai interface penguna, dilakukan dengan beberapa tahapan, yaitu analisis Sistem, desain arsitekture sistem, perancangan perangkat keras, perancangan perangkat lunak , implementasi sistem dan pengujian sistem.



Gambar 1 Alur Penelitian

2.1 Analisis Sistem

Dari analisis masalah, maka dibuat server yang telah di konfigurasi, untuk menjadi web server, menyimpan aplikasi berbasis website sebagai interface penguna dan protocol MQTT berfungsi untuk jembatan komunikasi antara node perangkat IoT. Protokol MQTT adalah protokol komunikasi SUBSCRIBE dan PUBLISH, di mana sebelum mengirim dan menerima pesan client harus terlebih dahulu terkoneksi dengan koneksi TCP ke server broker MQTT yang telah di konfigurasi, selanjutnya publisher dan subscriber harus memiliki topik yang sama untuk saling komunikasi.

ISSN (Cetak) 2527-6042
eISSN (Online) 2527-6050

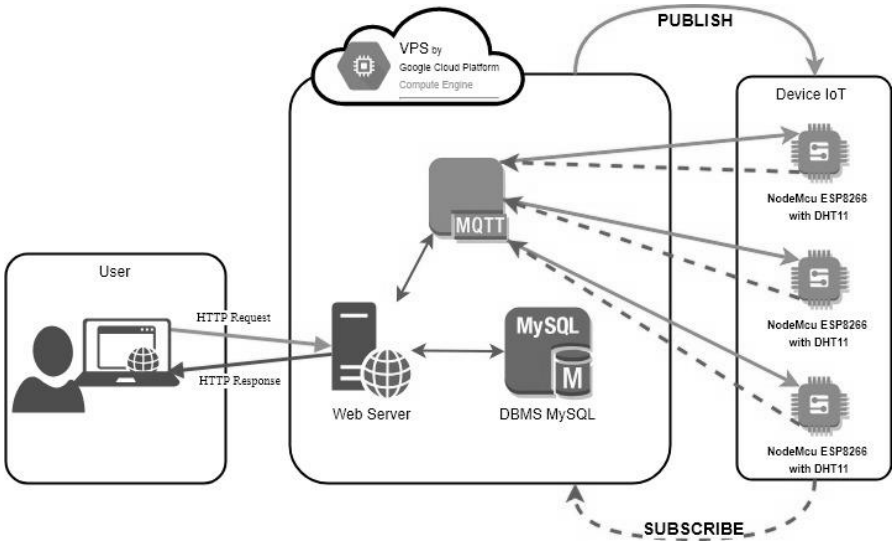
Publisher akan mengirim data ke server broker MQTT lalu broker akan mengirim ke subscriber, untuk identifikasi antara publisher dan subscriber harus memiliki topik yang sama agar pesan tersampaikan contoh topik "rumah/dapur/suhu" jadi setiap publisher dan subscriber akan memiliki topik yang sama.

Melakukan *over the air update firmware* kepada perangkat IoT di lakukan dengan cara mengambil *file firmware* hasil *compile* dari Arduino ide dengan *ekstensi .bin*, selanjutnya melakukan *upload* ke aplikasi berbasis website dan PUBLISH ke broker MQTT, lalu perangkat IoT melakukan SUBSCRIBE dan menerima *file ekstensi .bin* selajuntnya melakukan *update*.

Aplikasi berbasis website dibuat dengan PHP Framework Codeigniter, penguna harus *login* terlebih dulu, setelah *login* akan tampil pilihan untuk *upload file firmware ekstensi .bin, form publish* topik tujuan, dan *button publish*.

2.2 Desain Arsitektur Sistem

Arsitektur sistem penerapan protokol MQTT untuk *over the air update firmware*, mengunakan aplikasi berbasis website sebagai *interface* penguna dengan rancangan topologi sebagai berikut:



Gambar 2 Arsitektur system protokol MQTT untuk OTA update Firmware perangkat IoT

Dari rancangan topologi Gambar 2, terdiri dari *Vitual Private Server* (VPS) yang dikonfigurasi dengan protokol MQTT, Web Server Apache2, dan DBMS MySQL. Dalam penelitian ini perangkat IoT yang digunakan adalah modul NodeMCU ESP8266-12E sebagai sensor node untuk sensing data suhu dan kelembapan dengan DHT 11, sensor node tersebut akan menerima pembaruan *firmware*, dan penguna yang akan mengakses aplikasi berbasis website melalui browser.

Agar perangkat IoT dapat terhubung ke VPS maka harus di hubungkan dengan Wi-Fi yang sudah terhubung ke internet, seperti itu juga dengan aplikasi berbasis website yang akan di akses oleh penguna melalui browser internet, harus di pastikan memiliki nama domain atau ip publik yang bisa diakses melalui browser internet.

Agar rancangan arsitektur sistem gambar 2 dapat laksanakan sesuai dengan rancangan yang telah dibuat maka dibutuhkan hardware dan software sebagai berikut:

Tabel 1 Software dan Hardware sistem OTA dengan MQTT

Software	Hardware
Penguna	Laptop
Windows 10	Kabel USB 2.0 to Micro USB
Browser	
PuTTy	

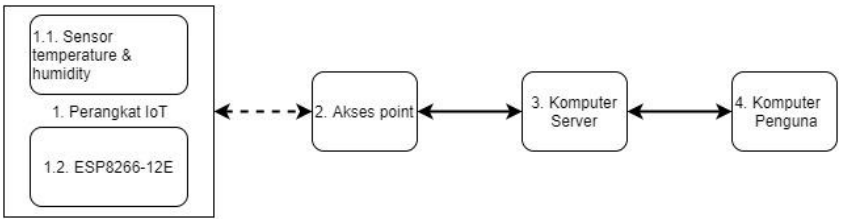
ISSN (Cetak) 2527-6042
eISSN (Online) 2527-6050

Atom IDE	
Server VPS	
LAMP (Linux, Apache,MySQL,PHP)	CPU 1 Ghz
Linux Ubuntu Server 16.04	Ram 1 GB
Broker EMQTT 2.0	Hardisk 10 GB
Perangkat IoT	
Arduino IDE	Nodemcu ESP8266 12E
	Sensor DHT 1

Dari Tabel 1 hardware dan software terbagi menjadi 3 jenis yaitu untuk Penguna, Virtual Private Server, dan perangkat IoT.

2.3 Rancangan Perangkat Keras

Dari rancangan arsitektur *sistem over the air update firmware* dengan protokol MQTT mengunakan aplikasi berbasis website sebagai interface penguna. Rancangan hardware terdiri dari perangkat IoT, akses point, komputer server, dan komputer penguna seperti yang terlihat pada gambar 3 dibawah ini.



Gambar 3 Rancangan hardware protocol MQTT untuk OTA update firmware perangkat IoT

Keterangan:

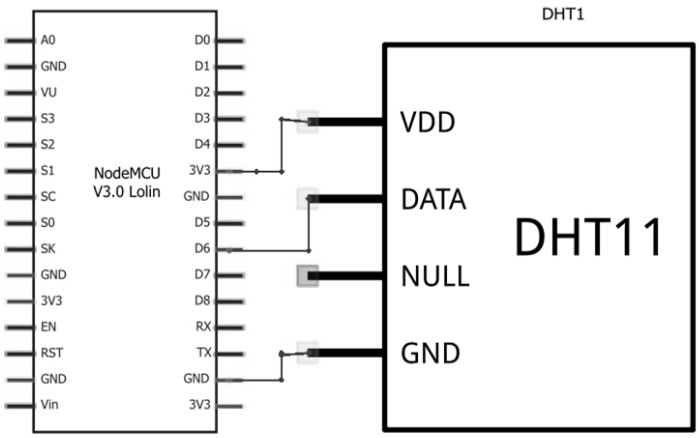
◄- - ► : Terhubung dengan jaringan *Wi-Fi*

◄——► : Terhubung ke jaringan internet

1. Perangkat IoT terdariri dari mikrokontroler dan sensor. Sensor temperature dan humadity mengunakan DHT 11 dan Mikrokontroler mengunakan NodeMcu ESP8266-12E.
2. Akses Point yang sudah terhubung ke internet.
3. Komputer Server sebagai tempat menginstall web server Linux, Apache, MYSQL, dan PHP, dan menginstall EMQTT Broker.
4. Komputer Penguna untuk penguna melakukan akses ke aplikasi berbasis website melalui browser untuk melakukan OTA dengan MQTT.

Perangkat IoT terdiri dari NodeMcu ESP8266-12E dan sensor DHT 11. Nodemcu ESP8266-12E digunakan untuk menerima *update firmware* dari aplikasi berbasis website yang kirim oleh penguna melalui protokol MQTT dan Sensor DHT 11 digunakan sebagai monitoring apakah program pada *firmware* yang di kirim berfungsi atau tidak, selain itu juga sensor di gunakan untuk mengirim data temperatur dan humadity di sekitar perangkat IoT dikirim ke aplikasi berbasis website. Jadi akan dibuat dua firmware yang telah terisi program PUBLISH data sensor, dan SUBSCRIBE topik *firmware* yang akan digunakan untuk menerima *update* dari aplikasi berbasis website, dengan rancangan skematik sebagai berikut:

ISSN (Cetak) 2527-6042
eISSN (Online) 2527-6050

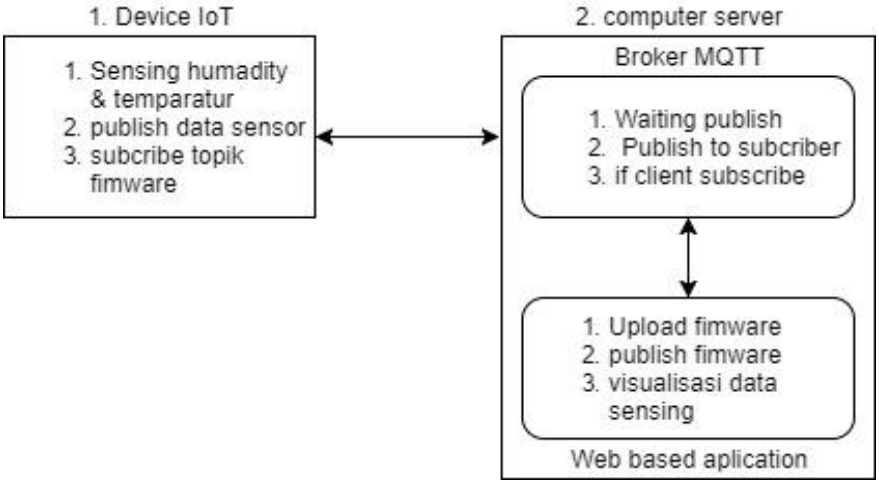


Gambar 1 Rangkaian Skematik Perangkat IoT

Dari rangkaian skematik gambar 4 DHT 11 memiliki 4 pin yaitu *VCC*, data, dan *GND*, dihubungkan ke Nodemcu ESP8266 12-E VCC dihubungkan ke VCC dan GND dihubungkan ke GND, data di hubungkan ke D6 pada NodeMcu ESP8266 12-E.

2.4 Rancangan Perangkat Lunak

Perancangan perangkat lunak untuk setiap komponen pada sistem Over The Air update firmware dengan protokol MQTT mengunakan aplikasi berbasis website, yang akan diaplikasakan pada perangkat IoT, dan server. Berikut adalah rancangan proses perangkat lunak pada Gambar 3.6



Gambar 2 Perancangan Proses Perangkat Lunak

Keterangan:

1. Desain proses perangkat lunak pada perangkat IoT
2. Desain proses perangkat lunak pada Komputer Server

⟷ Jaringan Internet

Dari desian perancangan proses perangkat lunak pada gambar 5 perangkat lunak di bagi menjadi 2 bagian perangkat lunak pada perangat IoT dan pada server. Perangkat lunak pada perangkat IoT melakukan s*ensing* data sensor suhu dan kelembapan lalu melakukan PUBLISH data sensor dan melakukan SUBSCRIBE topik *firmware* untuk menerima data *firmware* dari PUBLISH yang di lakukan aplikasi berbasis website. Pada *server broker* MQTT menunggu PUBLISH dari penguna jika ada yang melakukan PUBLISH dari penguna maka *broker* akan melakukan PUBLISH kepada penguna yang melakukan SUBSCRIBE dan memiliki topik yang sama, pada aplikasi berbasis website melakukan

ISSN (Cetak) 2527-6042
eISSN (Online) 2527-6050

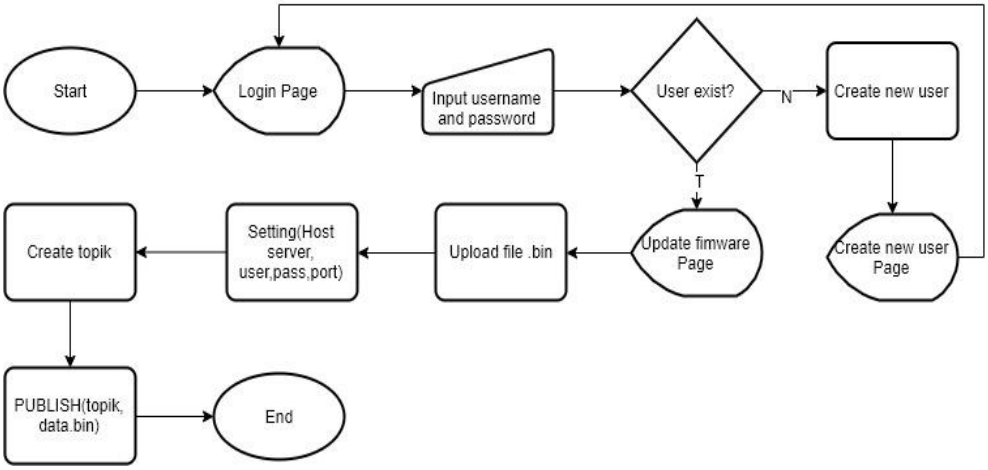
SUBSCRIBE data sensor dari perangkat IoT dan melakukan *upload firmware* lalu melakukan PUBLISH *file bin* ke broker MQTT, lalu meneruskan ke perangkat IoT yang memiliki topik sama.

2.1.1 Perangkat Lunak IoT

Perangkat lunak pada perangkat IoT memiliki dua tugas utama yaitu PUBLISH data hasil *sensing* dari sensor DHT 11 dan SUBSCRIBE topik untuk melakukan *update firmware* perangkat IoT. Sebelum melakukan SUBSCRIBE dan PUBLISH data pada perangkat IoT mengunakan *mikrokontroler* Nodemcu ESP8266-12E dimana telah memiliki modul Wi-Fi untuk melakukan koneksi ke *akses point* yang telah terhubung ke internet. Sebelum melakukan PUBLISH, perangkat IoT harus terkoneksi terlebih dahulu ke internet dan melakukan koneksi ke broker MQTT baru bisa melakukan update firmware dan PUBLISH data hasil *sensing* sensor DHT 11.

2.1.2 Rancangan Aplikasi Berbasis Website

Aplikasi berbasis website dibagi menjadi 3 halaman yang pertama adalah halaman *dashboard*, kedua halaman *update firmware*, dan halaman *setting password*. Proses PUBLISH *firmware* ke perangkat IoT, di mulai dari *upload file firmware*, *setting host* dan *port server* MQTT.



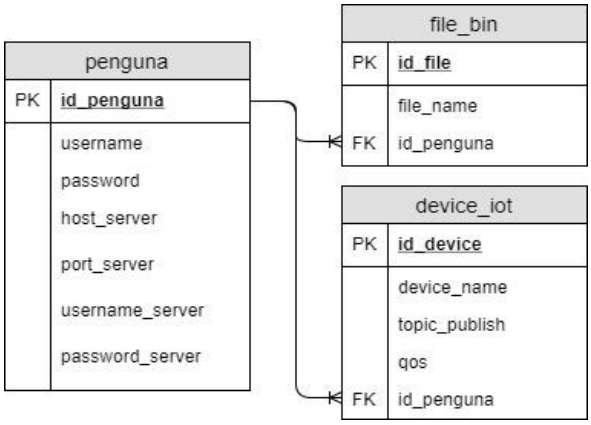
Gambar 3 Flowchare Aplikasi Berbasis Website

Dari Gambar 6 untuk melakukan PUBLISH *firmware*, penguna harus *login* terlebih dahulu dengan memasukan *username* dan *password*, jika *user* terdaftar maka akan langsung tampil halaman *update firmware* yang di dalamnya terdapat *form upload file.bin*, *setting* (*host server*, *user*, *pass*) dan *port*. jika akses *username* dan *password* ditolak maka penguna harus melakukan *register* mengisi *username* dan *password*.

2.4.3.1 Desain Database

Over the air update firmware dengan protocol MQTT mengunakan DBMS MySQL sebagai media penyimpanan yang digunakan penguna, dan perangkat IoT untuk menyimpan data, dengan design database sebagai berikut:

ISSN (Cetak) 2527-6042
eISSN (Online) 2527-6050



Gambar 7 Rancangan database OTA update firmware dengan MQTT

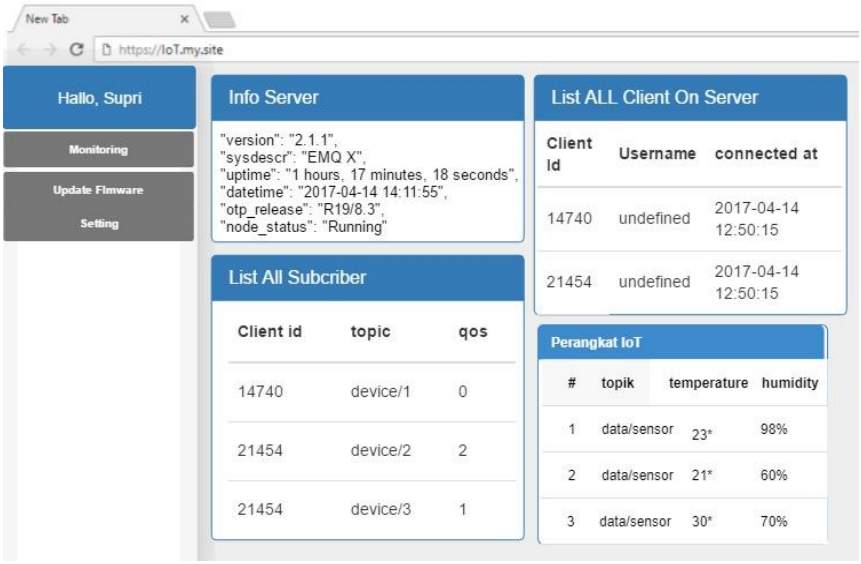
Dari design database Gambar 7, terdiri dari tiga tabel, yaitu tabel penguna, device_IoT dan file_bin. Fungsi tabel penguna untuk *verifikasi password login* pada aplikasi berbasis website, fungsi table *device* IoT untuk identifikasi nama perangkat IoT, *topic*, dan QoS, dan table *file bin* untuk menyimpan nama *file* yang di *upload* penguna dan menyimpan id penguna yang memiliki *file* tersebut.

2.4.3.2 Perancangan User Interface

Aplikasi berbasis website di gunakan untuk *update firmware* perangkat IoT, monitoring status *update* dan melihat apakah perangkat IoT sudah aktif atau belum baru selanjutnya melakukan *update*. Adapun rancangan *user interface* penguna terdiri dari *form login*, *dashboard* monitoring, *update firmware* dan *form setting*.

1. Mockup monitoring

Mockup monitoring terdiri dari informasi server, list client on server, list subscriber, dan data perangkat IoT.

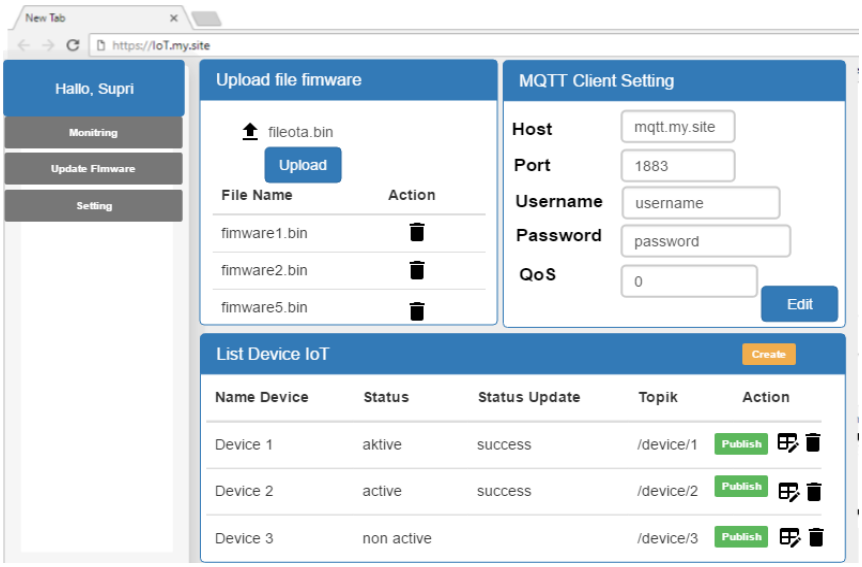


Gambar 8 *Mockup Monitoring*

Dari Gambar pada 8 informasi *server* berfungsi untuk melihat status broker *running* atau tidak, *list subscriber* untuk melihat id_client dan nama *topic* yang terkoneksi ke broker MQTT dan perangkat IoT berisi data dari *sensing temperature* dan *humadity*.

2. Mockup Update Firmware

Mockup update firmware terdiri dari *upload file firmware* , setting *client* MQTT, dan *list device* IoT.



Gambar 9 Mockup Update Firmware

Dari gambar 9 *upload file firmware* terdapat *button upload firmware* dan hapus. Pada *form* MQTT *client setting* terdapat *form* untuk *update host, port, username* dan lain-lain, digunakan untuk koneksi *client* ke broker MQTT. Pada list *device* adalah *form* untuk membuat *topic* dan mempublish *topic* ke broker.

2.5 Implementasi Sistem

Impelementasi sistem dilakukan berdasarkan desain sistem yang telah dirancang. Implementasi sistem terdiri dari implementasi perangkat keras dan implementasi perangkat lunak.

2.5.1 Implementasi Perangkat Keras IoT

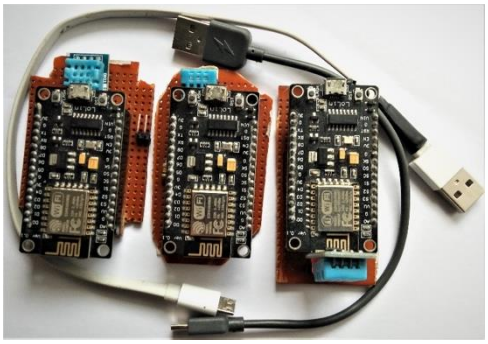
Untuk dapat melakukan implementasi perangkat IoT dibutuhkan alat dan bahan sebagai berikut:

Tabel 1 Alat dan Bahan Perangkat IoT

Nama alat dan bahan	Jumlah
Mikrokontroler Nodemcu ESP8266-12E	3
Sensor DHT 11	3
Kabel usb micro	3

Dari alat dan bahan pada table 2, dibuat hasil perancangan perangkat IoT sebagai berikut:

ISSN (Cetak) 2527-6042
eISSN (Online) 2527-6050



Gambar 4 Rangkaian Perangkat Keras IoT dan USB Micro

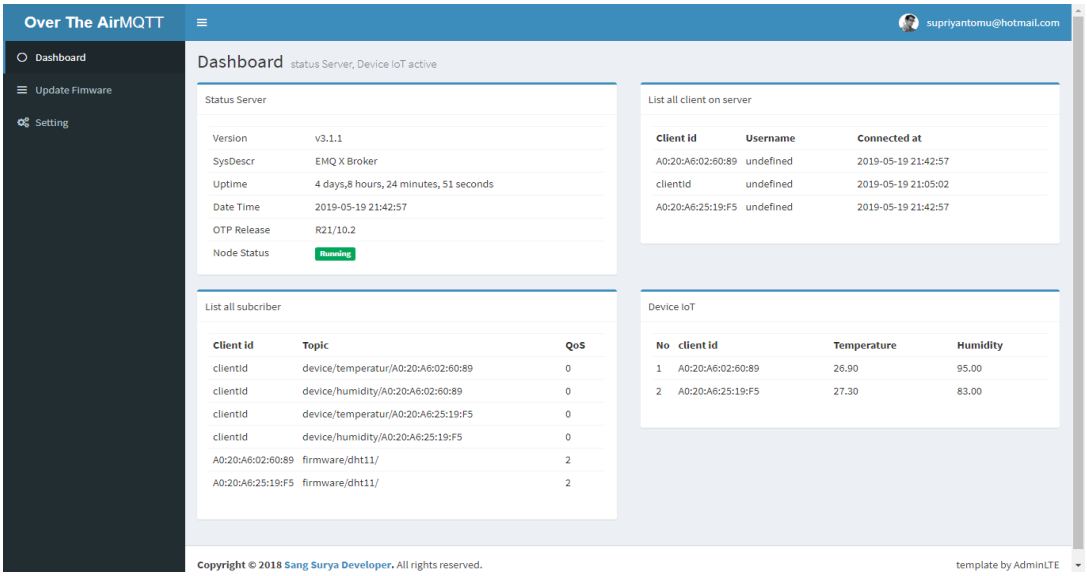
Dari hasil implementasi perangkat IoT pada gambar 6 terdapat 3 buah Nodemcu ESP8266-12E dengan masing-masing Nodemcu terhubung dengan sensor DHT 11. Setiap perangkat IoT yang telah dilakukan implementasi selanjutnya di program melalui kabel *usb to usb micro.*

3.5.6 Implementasi Perangkat Lunak

Implementasi perangkat lunak dibagi menjadi dua bagian implementasi perangkat lunak pada perangkat IoT dan implementasi perangkat lunak pada server. Perangkat lunak pada perangkat IoT adalah proses melakukan pemprogaman dengan bahasa C pada text editor arduino ide setelah di lakukan pemprogaman selanjutnya di lakukan *compile*, hasil dari *compile* berupa *file.bin* yang akan di *upload* pada aplikasi berbsis website. Implementasi perangkat lunak pada *server* terbagi menjadi tiga, pertama implementasi aplikasi berbasis website melakuka pemprogaman dengan bahasa program PHP fremwork codeigniter, kedua instalasi, konfigurasi dan setting Linux Aapache Mysql DAN PHP pada server, dan ketiga instalasi ,konfigirasi dan *setting* EMQX broker protokol MQTT.

3.5.6.1 Implementasi Aplikasi Berbasis Website

Halaman dashboard berisi informasi dari status *server*, jumlah *client* pada *serve*r, jumlah *subscriber*, dan informasi *temperature* dan *humadity* pada setiap *client*, Seperti terlihat pada Gambar 11



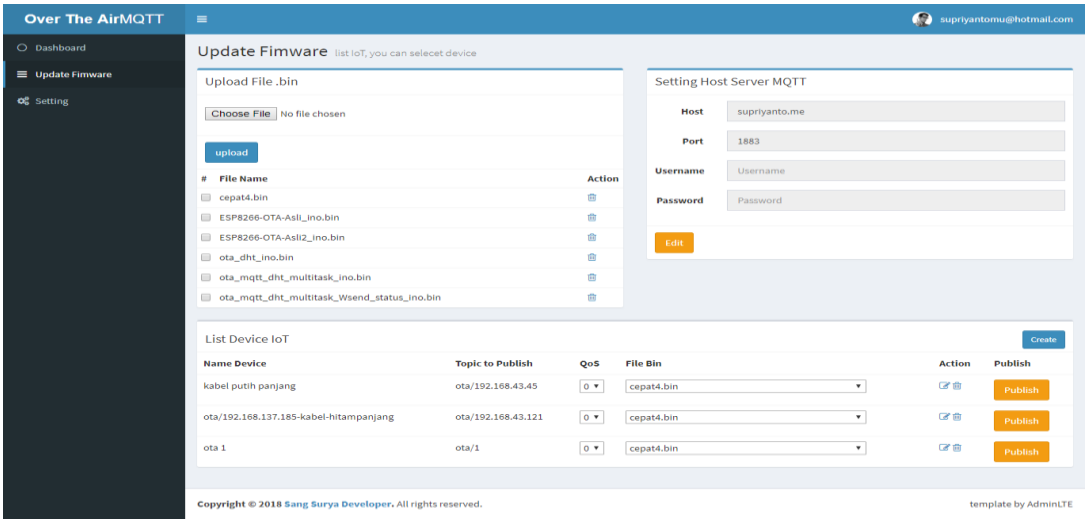
Gambar 11 Halaman Dashboard

Dari gambar 11 halaman *dashboard, status server* berisi informasi *broker* MQTT mulai dari versi yang digunakan, status *server* MQTT berjalan *uptime*, dan status *server running* atau tidak, *list all client*

ISSN (Cetak) 2527-6042
eISSN (Online) 2527-6050

on server berisi berapa jumlah *client* yang terkoneksi ke server MQTT berupa *client id*, dan waktu koneksi *client* sejak kapan, *list all subscriber* berisi semua *subscriber* berserta *topic* dan QoS yang digunakan dan *device* IoT berisi *client id*, *temperature* dan *humidty*, dari setiap perangkat IoT yang aktif.

Update firmware dilakukan dengan alur proses upload file. bin, lalu melakukan setting host dan port yang mengarah pada broker MQTT, selanjutnya membuat topik yang digunakan untuk *publish file bin*, dengan hasil implementasi pada gambar 12 di bawah ini:



Gambar 5 Halaman Update Firmware

Dari Gambar 12 adalah *form* untuk melakukan proses *publish firmware* mulai dari *upload file .bin*, *setting host server* MQTT, dan *list* topik, menentukan QoS, menentukan *file bin* yang akan di *publish* dan *button publish file .bin*. Fungis *upload file. bin* dilakukan dengan cara mengambil *file .bin* yang telah di *compile* pada gambar 4.6 , lalu pada *form upload file .bin* pada gambar 12 klik *button choose file*, pilih *file firmware.bin*, lalu klik *button upload*, yang akan ter *upload* dan disimpan dengan nama *file firmware.bin*.

2.6 Pengujian Sistem

Pengujian proses fungsional melakukan *over the air update firmware* pada aplikasi berbasis website dan menampilkan data hasil sensing.

Tabel 3 Daftar Fitur Uji Coba

No	Fitur	Keterangan
1	Login aplikasi berbasis website	Login dengan username dan password
2	Upload file . bin	Upload file .bin dari hasil build dari arduino ide
3	Setting host server MQTT	Setting domain, port, username dan password untuk akses server MQTT
4	Membuat topic	Membuan nama topic untuk publish ke fimware
5	publish file .bin	Memilih file .bin yangakan di publish klik button publish
6	Menampilkan data sensor	Menampilkan data sensor hasil sensing sensor

ISSN (Cetak) 2527-6042
eISSN (Online) 2527-6050

3. Hasil Penelitian Dan Pembahasan

Dari hasil pengujian fungsional berdasarkan tabel 3 daftar uji coba dilakukan dengan melihat apakah aplikasi berbasis website telah menghasilkan output yang diharapkan. Dengan hasil pengujian fungsional sebagai berikut:

No	Fitur	Keterangan	Berfungsi
1	Login aplikasi berbasis website	Login dengan username dan password	Ya
2	Upload file . bin	Upload file .bin dari hasil build dari arduino ide	Ya
3	Setting host server MQTT	Setting domain, port, username dan password untuk akses server MQTT	Ya
4	Membuat topic	Membuan nama topic untuk publish ke fimware	Ya
5	Publish file .bin	Memilih file .bin yangakan di publish klik button publish	Ya
6	Menampilkan data sensor	Menampilkan data sensor hasil sensing sensor	Ya

Dari tabel 3 fitur yang telah dibuat pada aplikasi berbasis website berfungsi sesuai dengan yang diharapkan.

4. Kesimpulan

Setelah melakukan implementasi dan pengujian pada bab IV dari tugas akhir dengan judul Rancang Bangun Layanan Over The Air Update Firmware dengan Protokol Message Queue Telemetry Transport (MQTT) pada IoT maka didapatkan hasil dan kesimpulan sebagai berikut.

1. Impelementasi protokol MQTT untuk melakukan over the air update firmware perangkat IoT berhasil dilakukan.
2. Pengunaan aplikasi berbasis website sebagai media interface interaksi penguna untuk melakukan upload file bin firmware perangkat IoT, lalu melakukan PUBLISH mengunaakan protokol MQTT ke perangkat IoT yang melakukan SUBSCRIBE berhasil dilakukan.

Refrensi

[1] Lund D, Morales M. *Worldwide and Regional Internet of Things* (IoT) 2014 – 2020 Forecast : A Virtuous Circle of Proven Value and Demand. IDC Anal Futur. 2014;(May):29.

[2] Evans D. *The Internet of Things How the Next Evolution of the Internet Is Changing Everything* [Internet]. 2011. Available from: https://www.cisco.com/c/dam/en_us/about/ac79/docs/innov/IoT_IBSG_0411FINAL.pdf

[3] Reißmann S, Pape C. *An Over the Air Update Mechanism for ESP8266 Microcontrollers.* ICSNC 2017 Twelfth Int Conf Syst Networks Commun An [Internet]. 2017;(October):11–7. Available from:https://www.researchgate.net/publication/320335879_An_Over_the_Air_Update_Mechanism_for_ESP8266_Microcontrollers

[4] Lee Jeffrey. *Over-The-Air Firmware: The Critical Driver of IoT* Success - DZone IoT [Internet]. https://dzone.com. 2017 [cited 2018 Mar 30]. Available from: https://dzone.com/articles/over-the-air-firmware-the-critical-driver-of-iot-s

[5] *ESP8266. OTA Update · ESP8266 Arduino Core* [Internet]. [cited 2018 Mar 29]. Available from:

ISSN (Cetak) 2527-6042
eISSN (Online) 2527-6050

http://esp8266.github.io/Arduino/versions/2.0.0/doc/ota_updates/ota_updates.html

[6] Quadri ASA, Sidek B. O. *An Introduction to Over-the-Air Programming in Wireless Sensor Networks*. Int J Comput Sci Netw Solut [Internet]. 2014;2:33–49. Available from: https://www.researchgate.net/publication/262181994%0AAn

[7] R A Atmoko*, R Riantini MKH. *IoT real time data acquisition using MQTT protocol*. Int Conf Phys Instrum Adv Mater. 2016;012003.

[8] Stanford Clark Andy NA. MQTT [Internet]. IBM. 1999 [cited 2017 Oct 9]. Available from: http://mqtt.org/

[9] Zhang Lucy. *Building Facebook Messenger* [Internet]. 2011 [cited 2017 Oct 10]. Available from: https://www.facebook.com/notes/facebook-engineering/building-facebook-messenger/10150259350998920